CMPS 2200 Assignment 2

Name: Mikey Sison

In this assignment we'll work on applying the methods we've learned to analyze recurrences, and also see their behavior in practice. As with previous assignments, some of of your answers will go in main.py.. You should feel free to edit this file with your answers; for handwritten work please scan your work and submit a PDF titled assignment-02.pdf and push to your github repository.

1. Derive asymptotic upper bounds of work for each recurrence below.

    a. $W(n) = 2W(n/3) + 1$.
    b. $W(n) = 5W(n/4) + n$.
    c. $W(n) = 7W(n/7) + n$.
    d. $W(n) = 9W(n/3) + n^2$.
    e. $W(n) = 8W(n/2) + n^3$.
    f. $W(n) = 49W(n/25) + n^{3/2} \log n$.
    g. $W(n) = W(n - 1) + 2$.
    h. $W(n) = W(n - 1) + n^c$, with $c \geq 1$.
    i. $W(n) = W(n) + 1$

Assignment 2

Question 1

1 a: $W(n) = 2W(n/3) + 1$

$C(Root) = 1$

$C(1) = 1(\frac{n}{3}) + 1 \cdot (\frac{n}{3}) = \frac{2n}{3}$

$C(2) = 1(\frac{n}{9}) + 1 \cdot (\frac{n}{9}) + 1 \cdot (\frac{n}{9}) + 1 \cdot (\frac{n}{9}) = \frac{4n}{9}$

$\frac{2n}{3} > \frac{4n}{9}$

$\therefore$ Root Dominated

$\boxed{a)\ O(n)}$

b) $W(n) = 5W(\frac{n}{4}) + n$

$C(Root) = n$

$C(1) = \frac{n}{4} + \frac{n}{4} + \frac{n}{4} + \frac{n}{4} + \frac{n}{4} = \frac{5n}{4}$

$C(2) = \frac{25n}{16}$

$\frac{25n}{16} > \frac{5n}{4}$

$\therefore$ Leaf Dominated

$\boxed{b)\ O(n^{\log_4(5)})}$

c) $W(n) = 7W(\frac{n}{7}) + n$

$C(Root) = n$

$C(1) = \frac{n}{7} + \frac{n}{7} + \frac{n}{7} + \frac{n}{7} + \frac{n}{7} + \frac{n}{7} + \frac{n}{7} = n$

$n = n \therefore$ Balanced

$\boxed{c)\ O(n \cdot \log_7(n))}$

d $W(n) = 9W(\frac{n}{3}) + n^2$

$C(Root) = n^2$

$C(1) = 9(\frac{n}{3})^2 = \frac{9n^2}{9}$ $\quad n^2 = n^2 = n^2 \therefore$ Balanced

$C(2) = 81(\frac{n}{9})^2 = \frac{81n^2}{81}$

$\boxed{O(n^2 \log(n))}$

e $W(n) = 8W(\frac{n}{2}) + n^3$

$C(Root) = n^3$

$C(1) = 8(\frac{n}{2})^3 = \frac{8n^3}{8}$ $\quad n^3 = n^3 = n^3 \therefore$ Balanced

$C(2) = 64(\frac{n}{2})^3 = \frac{64n^3}{2^{3 \cdot 3}} = \frac{64n^3}{64}$

☆ f $\boxed{O(n^3 \log(n))}$

g $W(n) = W(n-1) + 2 \quad = n+1$ $\quad$ Root Dominated

$= W((n-1)-1) + 2 \quad = n$ $\quad \boxed{O(n)}$

$W((n-1)-1-1) + 2 \quad = n-1$

$W(n-1)-1-1-1) + 2 \quad = n-2$

h $W(n) = W(n-1) + n^c$

$= n^c$ $\quad$ = Root Dominated

$= (n-1)^c$ $\quad \boxed{O(n^c)}$

$= (n-1-1)^c$

$= (n-3)^c$

i $W(n) = W(\sqrt{n}) + 1$

$= 1$ $\quad$ Leaf Dominated

$= \sqrt{n} + 1$ $\quad \boxed{O(\log \log n)}$

$= \sqrt{n} + 2$

Depth =

cost at level $= 1 \quad \therefore \quad 1 \cdot n = n$

☆ f) $49W(\frac{n}{25}) + n^{3/2} \log n$

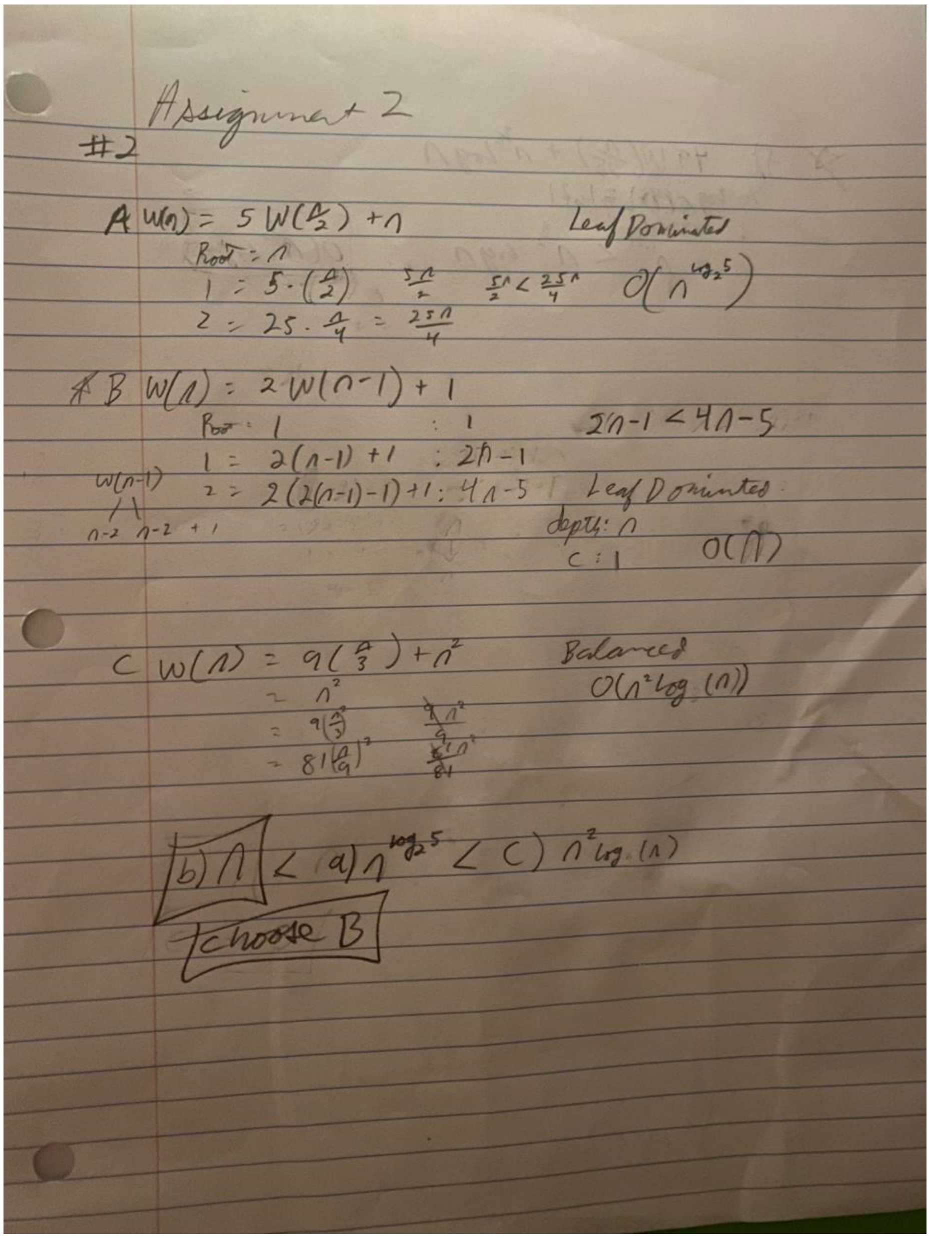
$\log_{25}(49) = 1.21$

$n^{1.21} < n^{3/2} \cdot \log n \quad \therefore \quad \boxed{O(n^{3/2} \log n)}$

2. Suppose that for a given task you are choosing between the following three algorithms:

- Algorithm *A* solves problems by dividing them into five subproblems of half the size, recursively solving each subproblem, and then combining the solutions in linear time.
- Algorithm *B* solves problems of size *n* by recursively solving two subproblems of size $n-1$ and then combining the solutions in constant time.
- Algorithm *C* solves problems of size *n* by dividing them into nine subproblems of size $n/3$, recursively solving each subproblem, and then combining the solutions in $O(n^2)$ time.

What are the asymptotic running times of each of these algorithms? Which algorithm would you choose?



Assignment 2

#2

A $W(n) = 5W(\frac{n}{2}) + n$ Leaf Dominated

Root: n

1: $5\cdot(\frac{n}{2})$ $\frac{5n}{2}$ $\frac{5n}{2} < \frac{25n}{4}$ $O(n^{\log_2 5})$

2: $25\cdot\frac{n}{4} = \frac{25n}{4}$

B $W(n) = 2W(n-1) + 1$

Root: 1 : 1 $2n-1 < 4n-5$

1 = $2(n-1)+1$: $2n-1$

2 = $2(2(n-1)-1)+1$: $4n-5$ Leaf Dominated

$W(n-1)$

$n-2$ $n-2$ + 1

depth: n $O(n)$

c: 1

C $W(n) = 9(\frac{n}{3}) + n^2$ Balanced

$O(n^2\log(n))$

= n^2

= $9(\frac{n}{3})^2$ $\frac{9n^2}{9}$

= $81(\frac{n}{9})^2$ $\frac{81n^2}{81}$

$\boxed{\text{b) } n} <$ a) $n^{\log_2 5} <$ c) $n^2\log(n)$

$\boxed{\text{Choose B}}$